

Programming Assignment 3: SDN and Ryu

COL 724

October 19, 2023

Goal: This assignment will give you hands-on experience with Ryu. You will learn how to implement specific network policies using OpenFlow-like APIs.

1 Setup

You will need to install Mininet and Ryu for this assignment.

Mininet

Refer to the instructions in Assignment 1 to install Mininet

Ryu

The installation instructions and tutorial for Ryu can be found here: https://ryu.readthedocs.io/en/latest/getting_started.html

Resources

Some useful commands:

- Ping between hosts h1 and h2: `h1 ping h2`
- Any command you want to send to host h1: `h1 cmd`
- Open up a new terminal for host h1: `xterm h1`. Make sure you set up x11 forwarding if you are on BaadalVM.
- Analyze network traffic: `tcpdump`
- Print the rules currently installed on switches: `dpctl dump-flows`
- Running Ryu apps: `ryu-manager [--VERBOSE] app.py`

2 Instructions

First, create the custom topology shown in Figure 1 using Mininet.

Part 1: Controller Hub and Learning Switch

You will compare the performance of a Hub Controller and a Learning Switch. A Hub Controller redirects all the traffic on a switch to itself and then forwards it to all switch ports except the incoming port. A Learning Switch installs flow rules on the switches based on the MAC to Port mappings it learns from incoming traffic. Begin by implementing a Controller Hub on both switches, S1 and S2. Following that, implement a Learning Switch. Answer the following questions:

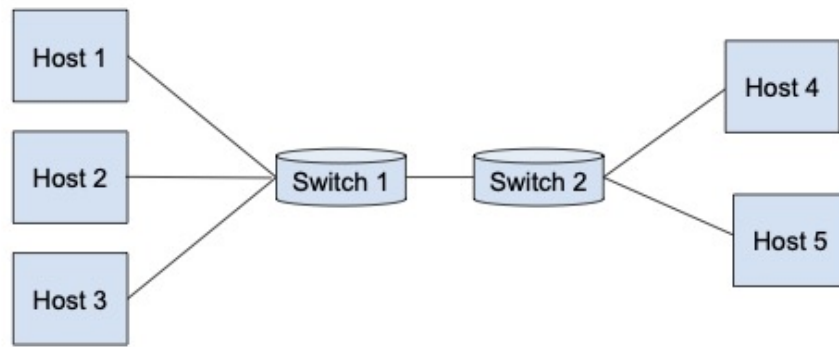


Figure 1: Custom Topology

- in both scenarios, conducting 3 pings for each case. Report the latency values. Explain the observed latency differences between the Hub Controller and Learning Switch. Also, explain differences (if any) observed between h2 and h5 for both controller types.
- Run a throughput test between h1 and h5. Report the observed values. Explain the differences between the Hub Controller and Learning Switch.
- Run pingall in both cases and report the installed rules on switches.

Hint: You can use the example code in the Ryu codebase.

Code Submission: You should submit two files, `learning_switch.py` and `controller_hub.py`. Each file should be a standalone Ryu app and should be able to run using the following command: `ryu-manager file_name.py`. The same goes for the next two parts.

Part 2: Firewall and Monitor

A network operator running the above topology needs to restrict communication between specific hosts because of security reasons. More specifically, the operator wants to prevent communication between H2 and H3 with H5, and H1 with H4. In addition, there is a requirement to count all packets coming from host H3 on switch S1. Implement a firewall+monitor that achieves the desired behavior.

The firewall rules can be installed when the switches register with the controller. The switches are learning switches as implemented in Part 1. Answer the following questions:

- Run pingall and report the results.
- Report the installed rules on both switches. Can you think of ways to minimize the number of firewall rules on the switch?
- Suppose the network operator intends to implement firewall policies in real time. How can she ensure that the pre-existing rules do not interfere with the firewall policy?

Code Submission: You should submit a single file called `firewall_monitor.py` containing both the firewall and monitoring functions.

Part 3: Load Balancer

Consider another scenario in which H4 and H5 function as servers. The network operator's objective is to employ a load balancer that evenly distributes requests from the remaining hosts to these servers in a round-robin fashion. These hosts connect to the servers via a virtual IP address, specifically 10.0.0.42. You need to implement such a load balancer on switch S1.

Hint: Pay attention to ARP Requests

The primary focus should be on crafting the appropriate flow rules. There's no need to be concerned with maintaining session persistence. Address the following question:

- Have the three hosts (H1, H2, and H3) ping the virtual IP and report the installed rule on the switches.
- If you were to implement a load balancing policy that considers the load on these servers, what additional steps would you take? [No need to implement it in the code, just describe the additional steps.]

Code Submission: You should submit a single file called `load_balancer.py`

Final Submission

Submit a single zip file comprising code and a report for all three parts.